

BSTZ No. 42P16548
Express Mail No.:EV323393887US

UNITED STATES PATENT APPLICATION

FOR

METHOD AND APPARATUS TO SUPPORT LEGACY MASTER BOOT RECORD
(MBR) PARTITIONS

Inventor:

Maresh S. Natu

Prepared by:

Blakely, Sokoloff, Taylor & Zafman LLP
12400 Wilshire Boulevard, Suite 700
Los Angeles, California 90025
(714) 557-3800

METHOD AND APPARATUS TO SUPPORT LEGACY MASTER BOOT RECORD
(MBR) PARTITIONS

FIELD

[0001] Embodiments of the invention relate generally to the field of computing systems, and more particularly, to a method and apparatus to support legacy master boot record (MBR) partitions in a storage device.

DESCRIPTION OF RELATED ART

[0002] By the use of interface device cards, today's computer systems are increasingly capable of supporting more technologically advanced input/output (I/O) devices, as well as a greater number of I/O devices. An interface device card may be utilized to couple the rest of the computer system to one or more I/O devices. Examples of these I/O devices include, but are not limited to, storage devices such as hard disk drives, floppy disk drives, Compact Disk Read Only Memory (CD ROM) drives, Digital Video Disk (DVD) drives, as well as monitors, keypads, modems, printers, etc. The interface device cards that support these I/O devices, as well as other types of devices, often support newer firmware interface standards as well as legacy standards. In order to support these standards, the interface device cards typically have Firmware Interface (FI) Option ROM (OPROM) and/or Legacy OPROM.

[0003] OPROM is typically firmware in the interface device card that is used to control a bootable peripheral I/O device. More particularly, OPROM includes pre-boot code that initializes a hardware device (e.g. an I/O device) that can be added to the computer system via the interface device card. The OPROM includes code to utilize the device.

[0004] Legacy OPROM is based on the original International Business Machine® (IBM) personal computer (PC)-Advanced Technology (AT) architecture, which is somewhat limited in its capabilities. For example, Legacy OPROM executes in 16-bit mode. On the other hand, newer firmware interface standards, implemented in FI OPROM, have advantages over legacy OPROM, such as being faster, more reliable, etc.

[0005] In the case where a peripheral I/O device is a storage device, such as a disk drive, two different schemes are typically used for addressing individual blocks of data on a disk of the disk drive. When Legacy Operating Systems (OSs) are utilized, which

42P16548

in turn utilize the Legacy OPROM of the interface device, a legacy Cylinder, Head, Sector (CHS) scheme is used. In the CHS scheme, a block of data can be identified and addressed by the starting and ending cylinder (C), head (H) of the disk drive, and Sector (S). The CHS scheme is an older scheme of disk access and as technology advances is being replaced by a second simpler Logical Block Addressing (LBA) scheme.

[0006] In contrast, the second LBA scheme uses a flat numbering scheme. The LBA scheme is typically used by newer firmware interfaces using FI OPROM of the interface device card. For example, the LBA scheme is a method used with small computer system interface (SCSI) disk drives and is also used with newer operating systems. Both of these two different schemes are typically used for addressing individual blocks of data on a disk of a hard disk drive.

[0007] For example, typically on the first sector of the disk of a hard disk drive, a data structure known as the master boot record (MBR) is present. The MBR is created when the disk is initially partitioned and is updated thereafter to reflect new disk partitions. Also, the MBR is executed when a computer boots up. When the MBR is executed it begins the boot process by looking at a partition table to determine which partition is to be used for booting. It then transfers program control to the boot sector of that partition, which continues the boot process.

[0008] The MBR contains a table of partitions and entries for each partition. Each partition has partition entries related to the address and size of the particular partition, as well as other data. Each partition may have data fields related to legacy CHS partition data and LBA partition data. Thus, for each partition entry, the partition entry may have data in the CHS scheme and/or the LBA scheme.

[0009] Accordingly, Legacy boot code associated with a Legacy operating system can utilize the particular Legacy partition address data (i.e. CHS data) to find and load the boot sector of the active partition and non-Legacy boot code (e.g. associated with a device under the control of a interface device using a newer FI standard) may utilize the LBA addressing scheme to use the particular LBA partition address data to find and load the boot sector of the active partition.

[0010] However, when an FI partition utility creates a partition on the disk it will only fill in the known fields in the FI partition address data field of the MBR of the disk for the particular partition. Particularly, it will only fill in the LBA address partition data for the particular partition. The LBA address is known since the FI software interfaces

42P16548

are based on LBA. Unfortunately, the CHS fields (i.e. the legacy CHS partition data) are left blank for that particular partition.

BRIEF DESCRIPTION OF THE DRAWINGS

[0011] Figure 1 shows a block diagram of an illustrative embodiment of an exemplary system configuration in which embodiments of the invention can be practiced.

[0012] Figure 2A is a diagram showing an example of one way in which data may be configured on a disk of a hard disk drive.

[0013] Figure 2B illustrates an example of a partition table of a master boot record (MBR) of a disk.

[0014] Figure 3 is a block diagram illustrating an overview of the principle components of a firmware interface (FI) in a computer system and its relation to platform hardware and OS software, according to one embodiment of the invention.

[0015] Figure 4 illustrates an example of a process to implement an FI partitioning, according to one embodiment of the present invention.

[0016] Figure 5 illustrates a process to support Legacy MBR partitions for use by Legacy operating systems (OSs), according to one embodiment of the present invention.

DETAILED DESCRIPTION

[0017] In the following description, the various embodiments of the invention will be described in detail. However, such details are included to facilitate understanding of the invention and to describe exemplary embodiments for employing the invention. Such details should not be used to limit the invention to the particular embodiments described because other variations and embodiments are possible while staying within the scope of the invention. Furthermore, although numerous details are set forth in order to provide a thorough understanding of the embodiments of the invention, it will be apparent to one skilled in the art that these specific details are not required in order to practice the embodiments of the invention. In other instances details such as, well-known methods, types of data, protocols, procedures, components, electrical structures and circuits, are not described in detail, or are shown in block diagram form, in order not to obscure the invention. Furthermore, embodiments of the invention will be described in particular embodiments but may be implemented in hardware, software, firmware, middleware, or a combination thereof.

[0018] With reference to Figure 1, Figure 1 shows a block diagram of an illustrative embodiment of an exemplary system configuration 100 in which embodiments of the invention can be practiced. The system configuration 100 includes at least one processor 101 such as a central processing unit (CPU), a chipset 102 which may include a memory control hub (MCH) and an Input/Output (I/O) control hub (ICH), a basic input/output system (BIOS) 105 that may be coupled to chipset 102, system memory devices 113, one or more interface devices 120, a network interface 134, a Universal Serial Bus (USB) interface 135, and other I/O interfaces 140. The system 100 may be any type of computing system. For example, the system 100 may be a server computer, personal computer, network computer, mainframe, hand-held device, etc.

[0019] For the purposes of the present specification, the term "processor" or "CPU" refers to any machine that is capable of executing a sequence of instructions and shall be taken to include, but not be limited to, general purpose microprocessors, special purpose microprocessors, application specific integrated circuits (ASIC), multi-media controllers, signal processors and microcontrollers, etc. In one embodiment, the CPU 101 is a general-purpose microprocessor that is capable of executing an Intel® Architecture instruction set. For example, the CPU 101 can be one of the PENTIUM

42P16548

classes of processors, one of the CELERON classes of processors, one of the TIANO classes of processors, or one of the ITANIUM classes of processors.

[0020] The CPU 101 may be coupled to the chipset 102 by a front-side bus (FSB). As previously discussed, in one example, the chipset 102 may include an MCH and an ICH, which are coupled to one another. The MCH may be responsible for servicing memory transactions that target the system memory devices 113. The MCH can be a stand-alone unit, an integrated part of a chipset, or a part of some larger unit that controls the interfaces between various system components and the system memory devices 113.

[0021] The system memory devices 113 can include any memory device adapted to store digital information, such as random access memory (RAM) 115 and read-only memory (ROM) 116. Examples of RAM include static random access memory (SRAM), dynamic random access memory (DRAM), synchronous dynamic random access memory (SDRAM), and double data rate (DDR) SDRAM or DRAM, etc. System memory devices 113 can also include non-volatile memory such as read-only memory (ROM) 116.

[0022] In one example, an ICH of the chipset 102 may provide interface control between the MCH and various I/O devices and ports such as peripheral component interconnect (PCI) slots and PCI agents via a PCI bus. Examples of such interfaces include a network interface 134 to interface the system 100 to a network (not shown), at least one USB port 135 that may be utilized to support USB devices such as a USB keyboard 136 and a USB mouse 138, and one or more other I/O interfaces to support various other types of I/O devices 142. I/O devices 142 include any I/O device to perform I/O functions. For example, I/O devices may include a monitor, a keypad, a modem, a printer, storage devices (e.g. Compact Disk ROM (CD ROM), Digital Video Disk (DVD), hard drive, floppy drive, etc.) or any other types of I/O devices, e.g., input devices (mouse, trackball, pointing device), media cards (e.g. audio, video, graphics), etc.

[0023] An interface device 120 having Firmware Interface (FI) Option ROM (OPROM) 122 and Legacy OPROM 124 may be utilized. Particularly, an interface device card 120 may be utilized to couple the rest of the system 100 to one or more devices. OPROM is typically firmware in an interface device card that is used to control bootable peripheral I/O devices. More particularly, OPROM includes pre-boot code that initializes hardware that can be added to the system 100 via an interface

42P16548

device card. Legacy OPROM 124 is based on the original IBM® PC-AT architecture and is somewhat limited in its capabilities. For example, Legacy OPROM 124 executes in 16-bit mode. On the other hand, the FI OPROM 122 has advantages over Legacy OPROM 124, such as being faster, more reliable, etc. In one embodiment, the FI OPROM may be compatible with Intel's Extensible Firmware Interface Standard (Extensible Firmware Interface Specification, Version 1.10, published December 1st, 2002) to provide an EFI interface, which likewise has many advantages over the Legacy standard. Whether Legacy OPROM or FI OPROM is utilized, the system BIOS 105 typically interrogates the OPROM(s) to determine which devices can be booted.

[0024] In one example, the interface device 120 may be a small computer system interface (SCSI) card. As is known, SCSI is a high-speed parallel interface standard to couple one or more devices to a computer system. Thus, coupled to the interface device may be one or more I/O devices, such as storage devices 126a,b, etc. For example, one of the storage devices 126 may be a hard drive. It should be appreciated that any sort of I/O device can be coupled through the interface device 120 to the rest of the system 100.

[0025] When a SCSI interface device card 120 is utilized using Legacy OPROM 124, the Legacy OPROM will initialize the SCSI hard drive 126 and provide Legacy INT 13 services that allow the rest of the system to access the SCSI hard drive 126. As is known, INT 13 is a DOS interrupt used to activate disk functions such as seek, read, write, format, etc. Similarly, when a SCSI interface device card 120 is utilized using FI OPROM, the FI OPROM will likewise initialize the SCSI hard drive 126 to allow the rest of the system to access the SCSI hard drive.

[0026] It should be appreciated that the Figure 1 system configuration 100 is only an exemplary illustration of an environment in which embodiments of the invention can be practiced. Further, those skilled in the art will recognize that the exemplary environment illustrated in Figure 1 is not intended to limit the embodiments of the invention.

[0027] Turning now to Figure 2A, Figure 2A is a diagram showing an example of one way in which data may be configured on a disk 200 of a disk drive, such as a hard disk drive 126. As shown in Figure 2A, a disk 200 may be divided into multiple sectors (e.g. sector-1 210, sector-2 212, etc.) and multiple cylinders (e.g. cylinder-1 216, cylinder-2 218, etc.). Sometimes the term track is used instead of the term cylinder.

42P16548

[0028] Two different schemes are typically used for addressing individual blocks of data on a disk 200 of a disk drive. These are the CHS scheme and the Logical Block Addressing (LBA) scheme. In the first scheme, the CHS scheme, the block address for the data to be addressed is represented in terms of three numbers C (Cylinder), H (Head) and S (Sector). In the CHS scheme, a block of data can be identified by the starting and ending cylinder (C), head (H) of the disk drive, and sector (S). The CHS scheme is an older scheme of disk access and as technology advances is being replaced by the simpler LBA scheme.

[0029] In contrast, the second LBA scheme uses a flat numbering scheme. The LBA scheme is a method used with SCSI and IDE disk drives to translate the CHS specifications of the drive into addresses that can be used by an enhanced Basic I/O System (BIOS). These two different schemes are typically used for addressing individual blocks of data on a disk 200 of a hard disk drive.

[0030] Also, as shown in Figure 2A, typically on the first sector of the disk 200 is a data structure known as the master boot record (MBR) 202. The MBR is created when the disk 200 is partitioned. The MBR is executed when a computer boots up. When the MBR is executed it begins the boot process by looking at a partition table to determine which partition is to be used for booting. It then transfers program control to the boot sector of that partition, which continues the boot process. The MBR contains a table of partitions and entries for each partition. Each partition has partition entries related to the address and size of the particular partition, as well as other data.

[0031] Turning to Figure 2B, Figure 2B illustrates an example of a partition table 220 of an MBR 202 of a disk 200. As shown in Figure 2B, the partition table 220 includes a plurality of partition entries 228a, 228b, 228c, etc. for each partition (e.g. partition-1, partition-2, partition-3, etc.) on the disk. Associated with each partition entry, is Legacy (CHS) partition data 238a, 238b, 238c, etc. for each partition (e.g. CHS-1, CHS-2, CHS-3, etc.) and LBA partition data 248a, 248b, 248c, etc., for each partition (e.g. LBA-1, LBA-2, LBA-3, etc.). Thus, for each partition entry, the partition entry has corresponding data in the CHS scheme and the LBA scheme to represent the address and size of the partition in each particular scheme, respectively. Accordingly, Legacy boot code can utilize the particular Legacy partition data to find and load the boot sector of the active partition and non-Legacy boot code that utilizes the LBA addressing scheme can use the particular LBA partition data to find and load the boot

sector of the active partition. It should be noted that most modern operating systems (OSs) and firmware interfaces use the LBA fields.

[0032] With reference now to Figure 3, Figure 3 is a block diagram illustrating an overview of the principal components of a firmware interface (FI) in a computer system and its relation to platform hardware and OS software, according to one embodiment of the invention. Figure 3 illustrates the various components that are used to accomplish OS boot and FI OS boot. The firmware interface provides an interface between the operating system and the platform firmware of a computer system. Particularly, the interface may be in the form of data tables that contain platform-related information, and boot and run-time service calls that are available to the OS loader and the OS. Together, these provide a standard environment for booting both a standard OS and a FI OS. The OS and the platform firmware communicate information necessary to support the OS boot process. In this way, firmware associated with an interface device can be seamlessly integrated with a computer system to add new hardware to the system.

[0033] In one example the FI may be in accordance with Intel®'s Extensible Firmware Interface Specification (Extensible Firmware Interface Specification, version 1.10, published December 1st 2002).

[0034] As shown in Figure 3, the system includes an operating system 302 and a Legacy OS loader 304 and an FI OS loader 310. The system also includes a basic input/output system (BIOS) 306, which may be coupled to a non-volatile memory 308, such as a flash memory. Further, an FI application program interface (API) 312 including FI boot services 314, FI run-time services 316, and FI drivers 318 may be utilized to provide the requisite functionality to operate an interface device associated with the FI.

[0035] Platform hardware 320 may include a motherboard 322 for the computer system. For example, this may include a processor 101, a chipset 102, system memory devices 113, etc. (Figure 1). Platform hardware 320 may further include Option ROM (OPROM) 324. Particularly, this may include FI OPROM and Legacy OPROM. In one example, as discussed with reference to Figure 1, FI OPROM 122 and Legacy OPROM 124 may be present in interface device 120, such as a SCSI interface device card. Also, non-volatile memory 308 may be separate non-volatile memory or may be part of system memory devices 113.

42P16548

[0036] Platform hardware 320 also includes one or more storage devices, such as a hard disk drive 126 as previously discussed with reference to Figure 1. A storage device of platform hardware 320 may include at least one OS partition 340 and at least one FI system partition 330. Included within the FI system partition are drivers 332 and an FI OS loader 334. For example, the previously described configuration allows firmware for a device, such as a SCSI interface device card 120, to be seamlessly integrated with the rest of the system to enable the use of a device, such as a hard disk drive, to be used with the system.

[0037] When an FI application 312 creates an MBR disk partition, it typically does so utilizing only the LBA addressing scheme and only fills in LBA partition address data for the partition and not Legacy CHS partition address data for the partition. However, it would be beneficial for the Legacy operating system 302, which can only read and understand Legacy CHS address data, to be able to read the appropriate CHS fields for the FI system partition 330.

[0038] The platform hardware 320 typically includes FI OPROM and Legacy OPROM 324. Continuing with our previous example, a SCSI interface device card 120 may include FI OPROM 122 and Legacy OPROM 124 to control a hard disk drive 126. The system BIOS 306 will typically run one of the two OPROM during the boot process to boot the interface device card 120. If the Legacy OPROM is run, the Legacy OPROM can be queried to find out the mapping it uses.

[0039] However, FI OPROM typically does not utilize the CHS addressing scheme, but rather uses the LBA addressing scheme. Thus, when the FI OPROM is run the mapping information is not available in the CHS scheme for the Legacy operating system 302.

[0040] Particularly, when an FI partition utility creates a partition on the disk, it will fill in the known fields in the FI partition entry, in the MBR 202 of the disk 200 (Figure 2). Particularly, it will fill in the LBA partition data 248 for the particular partition 228. The LBA address is known since the FI software interfaces are based on the LBA. However, the CHS fields (e.g. Legacy CHS partition data 238) are left blank. The location of the FI disk partition in LBA is communicated to the system BIOS 306. Also, the system BIOS 306 stores this information in non-volatile memory 308. When the FI application 312 is done with the partitioning and the user wants to boot an operating system, the user will reboot the system.

42P16548

[0041] Before booting a Legacy operating system 302, the system BIOS 306 reads the non-volatile memory 308 to find out if there any disk partitions without CHS fields. If there are, the system BIOS 306 runs the Legacy OPROM 124 of the interface device 120 for the storage device 126 instead of the FI OPROM 122. Once the Legacy OPROM 124 is run, the CHS mapping can be retrieved by the system BIOS 306 and the necessary partition entries can be updated with corresponding CHS values in both the non-volatile memory 308 and in the storage device (e.g. a hard disk drive 126) itself. For example, Legacy CHS partition data 238 of the partition table 220 of the MBR 202 of the disk 200 of can be updated under the control of system BIOS 306. The Legacy OS 302 can now understand the partition entries in both the non-volatile memory 308 and the disk 200.

[0042] Turning now to Figure 4, Figure 4 illustrates an example of a process 400 to implement an FI partitioning. At block 402 the system BIOS 306 is run. The non-volatile memory 308 is read (block 404). Next, the FI OPROM is executed (block 406). The FI shell is booted at block 408 and at block 410 the disk partitioning utility is invoked. After the disk partitioning, wherein partitioning data related to the address and size of the FI system partition 330 is created and saved in the MBR in accordance with the LBA partitioning scheme (as previously discussed), this LBA partition data is also saved in non-volatile memory 308 as an FI variable for access by system BIOS 306 (block 412).

[0043] With reference now to Figure 5, Figure 5 illustrates a process 500 to support legacy MBR partitions for Legacy operating systems (OSs), according to one embodiment of the present invention. First, the Legacy OS 302 is booted (block 502). Next, at block 504, the system BIOS 306 is run. The non-volatile memory 308 is read by the system BIOS 306 (block 506). Then, at block 508, the process 500 under the control of the system BIOS 306 determines whether Legacy partition data (e.g. CHS data) is present for all of the disk partitions recorded in non-volatile memory. If CHS data for all the partitions already is already present, then standard processing proceeds (block 510).

[0044] However, if Legacy partition data (e.g. CHS data) does not exist for a particular partition, then at block 512, the Legacy OPROM is executed. Next, Legacy mapping data (e.g. CHS address data) is obtained from the Legacy OPROM for the partition such that the partition data utilizing the CHS addressing scheme is available for the

42P16548

partition (block 514). This CHS address data for the partition is stored in the non-volatile memory 308 for use by the Legacy OS the next time it is booted.

[0045] Further, at block 516, the necessary partition entries on the disk itself are updated. For example, looking at Figure 2B, for the particular partition entry 228, the corresponding Legacy CHS partition address data 238 is filled in such that both Legacy (e.g. CHS) partition address data 238 and LBA partition address data 248 are present for the partition. The Legacy OS is now booted. In this way, the Legacy OS may be able to read and understand the appropriate CHS address fields for an FI partition. Functional blocks 506 through 516, previously discussed, may be executed under control of the system BIOS 306.

[0046] It should be appreciated by those of skill in this art, that although the embodiments of the present invention have been illustrated utilizing as an example a SCSI interface card and a hard disk drive utilizing a disk having partition data, that embodiments of the invention can be utilized with any sort of interface device and device coupled thereto.

[0047] While embodiments of the present invention and its various functional components have been described in particular embodiments, it should be appreciated that the embodiments of the present invention can be implemented in hardware, software, firmware, middleware or a combination thereof and utilized in systems, subsystems, components, or sub-components thereof.

[0048] When implemented in software or firmware, the elements of the present invention are the instructions/code segments to perform the necessary tasks. The program or code segments can be stored in a machine readable medium (e.g. a processor readable medium or a computer program product), or transmitted by a computer data signal embodied in a carrier wave, or a signal modulated by a carrier, over a transmission medium or communication link. The machine-readable medium may include any medium that can store or transfer information in a form readable and executable by a machine (e.g. a processor, a computer, etc.). Examples of the machine-readable medium include an electronic circuit, a semiconductor memory device, a ROM, a flash memory, an erasable programmable ROM (EPROM), a floppy diskette, a compact disk CD-ROM, an optical disk, a hard disk, a fiber optic medium, a radio frequency (RF) link, etc. The computer data signal may include any signal that can propagate over a transmission medium such as electronic network channels, optical

42P16548

fibers, air, electromagnetic, RF links, bar codes, etc. The code segments may be downloaded via networks such as the Internet, Intranet, etc.

[0049] Further, while embodiments of the invention have been described with reference to illustrative embodiments, these descriptions are not intended to be construed in a limiting sense. Various modifications of the illustrative embodiments, as well as other embodiments of the invention, which are apparent to persons skilled in the art to which embodiments of the invention pertain, are deemed to lie within the spirit and scope of the invention.